این مار خوش خط و خال

قسمت ششم

آشنایی و کار با زبان برنامه‌نویسی پایتون

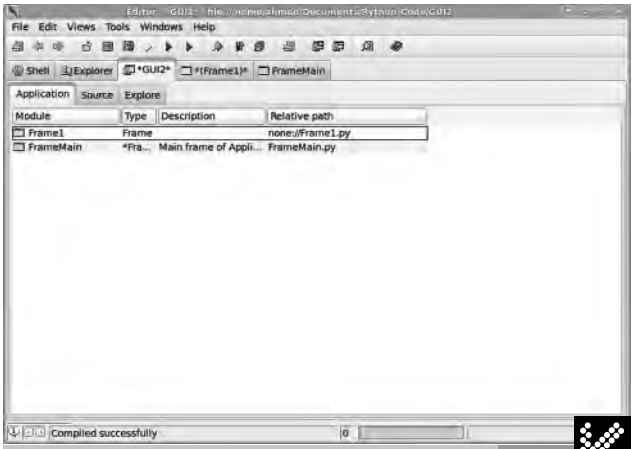
» احمد شریف پور

در مقاله قسمت قبلی برنامه‌ای ایجاد کردیم که با کلیک یک دکمه، یک پیغام را به نمایش در می‌آورد. امیدواریم که تا انتشار این شماره، فرصت کافی را برای کار با Boa-Constructor پیدا کرده باشید. در این شماره می‌خواهیم سیستم کار با برنامه‌هایی را یاد بگیریم که از چندین پنجره یا به اصطلاح فریم مجزا و مستقل استفاده می‌کنند. هر چند برنامه این قسمت تنها از دو فریم استفاده می‌کند و کار پیچیده‌ای انجام نمی‌دهد، اما اصول ساخت فریم‌های جدید و برقراری ارتباط بین آن‌ها را به شما آموزش خواهد داد.

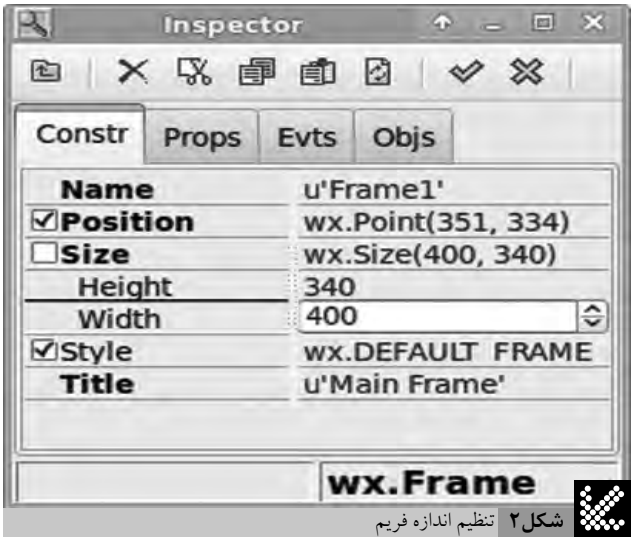
شروع

برنامه Boa Constructor را اجرا کنید و در قاب Editor تمام زبانه‌ها به‌جز Shell و اکسپلورر را با استفاده از ترکیب کلیدهای CTRL+W ببندید. با این کار مطمئن می‌شویم، کار را از پایه و روی فایل‌های تازه شروع کرده‌ایم. حال با کلیک دکمه wx.App یک پروژه جدید بسازید. در صورت نیاز به راهنمایی به مقاله شماره قبل مراجعه کنید. پیش از هر کار دیگری، Frame1 را با نام FrameMain.py و App1 را با نام Gui2.py ذخیره کنید. در حالی‌که زبانه GUI2 در قاب Editor انتخاب شده است، به نوار ابزارها مراجعه کنید و در زبانه New و با کلیک دکمه wx.Frame که درست در کنار wx.App قرار گرفته است، یک قاب یا پنجره دیگر به پروژه اضافه کنید. اطمینان حاصل کنید که زبانه Application در زیر ستون Module همانند شکل ۱ هر دو فریم را نمایش می‌دهد. حال به فریم جدید مراجعه کرده و آن را با نام FrameSecond.py ذخیره کنید. سپس FrameMain را در Designer باز کنید و از زبانه Container/Layout در قاب اصلی، یک wx.Panel به آن اضافه کنید. اندازه آن را طوری تنظیم کنید که تمام فریم را بپوشاند. حال باید برخی از خصوصیات فریم را دستکاری کنیم. در قاب Inspector و در زبانه Objs شیء Frame را انتخاب کنید و در زبانه Constr مقدار title را به Main Frame تبدیل کنید. در انتهای مقاله درباره قوانین نام‌گذاری صحبت خواهیم کرد. با کلیک علامت تیک آیتم Size اندازه فریم را به 400x340 تغییر دهید. با کلیک روی این علامت، لیستی باز می‌شود که در آن مقدار Height باید برابر ۳۴۰ و مقدار Width برابر چهارصد تنظیم شود (به شکل ۲ دقت کنید).

حال زبانه Props را کلیک کرده و مقدار خاصیت Centered را به wx.BOTH تنظیم کنید. دکمه Post را (شکل یک تیک که تغییرات انجام شده



شکل۱ افزودن فریم به برنامه پایتون



شکل۲ تنظیم اندازه فریم

شکل۳ فریم اصلی برنامه

را به فایل اصلی اعمال می‌کرد) کلیک کرده و کل پروژه را ذخیره کنید. حال با کلیک دکمه زردرنگ مثلثی برنامه را اجرا کنید. فریم اصلی برنامه ما باید درست در وسط صفحه نمایش و با عنوان Main Frame ظاهر شود. فعلاً برنامه را با کلیک دکمه Close در قسمت بالا و سمت راست پنجره ببندید.

دوباره فریم Main Frame را در Designer باز کنید. از زبانه Buttons قاب اصلی، دو دکمه wx.Button را به فریم اضافه کنید، به گونه‌ای که یکی در نزدیکی مرکز فریم و یکی بالای دیگری قرار بگیرند. دکمه بالایی را انتخاب کرده و با استفاده از قاب Inspector و زبانه Constr نام آن را به btnShowNew و خاصیت Label آن را به «Show the other frame» تنظیم کنید. با استفاده از ترکیب کلیدهای Shift و کلیدهای جهت‌نما، اندازه دکمه را به گونه‌ای تنظیم کنید که تمام نوشته Label آن قابل رؤیت باشد، سپس از ترکیب CTRL و کلیدهای جهت‌نما برای بازگرداندن آن به مرکز فریم استفاده کنید. حال دکمه پایینی را انتخاب کرده و همانند دکمه قبلی نام آن را btnExit و Label آن را برابر Exit تنظیم کنید. تغییرات را با دکمه Post (که آیکون آن شبیه یک تیک است) به فایل‌های محتوای کد ارسال کرده و برای مشاهده تغییرات برنامه را دوباره اجرا کنید. ظاهر برنامه شما باید همانند شکل ۳ باشد. اکنون برنامه را ببندید.

رویدادها

به احتمال برای شما هم تعجبی نداشته است که دکمه Exit هنوز کار نمی‌کند و عملیات بستن برنامه را باید با دکمه Close نوار عنوان برنامه انجام دهید. غالب برنامه‌های مبتنی بر GUI از سیستم هدایت رویداد (Event Driven) استفاده می‌کنند. در این روش، برنامه به‌صورت مداوم رویدادهای سیستم را نظیر کلیک‌ها، ضربه‌های کلید، حرکات ماوس، تغییر زمان و... بررسی می‌کند و اگر با رویدادی مواجه شود که برای آن عملیاتی تعریف شده باشد، آن عملیات را به اجرا در خواهد آورد، پس دلیل کار نکردن دکمه‌های برنامه ما این است که هنوز عملکردی را به رویداد کلیک آن‌ها نسبت نداده‌ایم. برای نسبت دادن عملکردهای مناسب به دکمه‌ها، فریم اصلی برنامه را در Designer باز کرده، سپس دکمه بالایی را انتخاب کرده و در قاب Inspector به زبانه Evts مراجعه کنید. روی ButtonEvent کلیک کرده، سپس روی wx.EVT_BUTTON دوبار کلیک کنید. دقت کنید که اکنون در قسمت زیرین عبارت OnBtnShowNewButton ظاهر خواهد شد (شکل۴). سپس این روال را برای دکمه Exit اجرا کرده و اطمینان حاصل کنید که عبارت OnBtnExitButton در قسمت زیرین بخش Evts قاب Inspector به نمایش درآمده است. تغییرات را Post کرده و فایل‌ها را ذخیره کنید. پس از آن به قاب Editor رفته و به قسمت انتهایی کد مربوط به FarmeMain مراجعه کنید.

در این‌صورت باید دو «متدی» را که به تازگی برای دکمه‌ها ساخته‌اید، ببینید. اکنون زمان آن رسیده که به سراغ فریم دوم برویم. فریم FramwSecond را که در ابتدای کار ساخته بودید، در Designer باز کرده و نام آن را به FrameSecond و عنوان آن را به Second Frame تنظیم کنید. به احتمال دیگر می‌دانید که خاصیت Name اسمی است که ما در کدهایمان برای ارجاع به اشیا از آن استفاده می‌کنیم و Label یا Title عنوانی است که روی شیء و برای کاربر نمایش داده می‌شود. اندازه فریم را به ۱۸۰×۴۰۰ و وضعیت Centered را روی wx.BOTH تنظیم کنید. یک دکمه wx.BUTTON به فرم اضافه کرده و آن را در وسط قسمت پایینی فریم قرار دهید. نام این دکمه را به btnFSExit و Label آن را به Exit تبدیل کنید و در نهایت مانند دکمه‌های قبلی، یک رویداد کلیک برای آن تعریف کنید. حال یک کنترل از نوع wx.StaticText به بخش بالایی فریم اضافه کرده و نام آن را به stHiThere و Label آن را به «Hi There . . . I›m the second frame !» تبدیل کنید. با مراجعه به زبانه Props و برداشتن تیک کنار Fonts فونت این برچسب متنی را به Sans با اندازه ۱۴ و Weight آن را به wxBOLD تنظیم کنید. حال این نوشته را درست به وسط قسمت بالایی فریم ببرید. برای این کار می‌توانید در قسمت Position از زبانه Constr با تنظیم مقادیر X و Y محل این متن را تعیین کنید. فریم شما باید همانند شکل ۵ باشد.

چسباندن اجزا

حال که ظاهر اجزای مختلف برنامه را آماده کرده‌ایم، زمان آن رسیده تا چسبی را که تمام این اجزا را به هم متصل می‌کند، به‌وجود آوریم. در قاب Editor روی زبانه GUI2 کلیک کنید و در قسمت زیرین زبانه Source

را انتخاب کنید. در زیر خطی که دستور

```
Import  FrameMain
```

را در خود دارد، خطی دیگر با دستور

```
import FrameSecond
```

اضافه کنید. تغییرات را ذخیره کرده و حالا زبانه FrameMain را فعال کنید و در زیر خط محتوای دستور

```
import wx
```

شما نیز خط زیر را اضافه کنید:

```
 import FrameSecond
```

اکنون کد را به سمت پایین پیمایش کنید تا خط زیر را بیابید.

```
def __init__(self,parent):
```

در زیر خط محتوای دستور self._init_ctrls(parent)، شما نیز خطی را با دستور زیر وارد کنید.

```
self.Fs=FrameSecond.FrameSecond(self)
```

حال در زیر بخش تعریف توابع کلیک، یعنی

```
def OnBtnShowNewButton(self,event)
```

با قرار دادن یک علامت # دستور event.Skip() را به کامنت تبدیل کنید و دو خط زیر را به ادامه آن بیافزایید.

```
self.Fs.Show()
self.Hide()
```

در انتها و در قسمت مربوط به متد OnBtnExitButton دوباره دستور event.Skip() را به کامنت تبدیل کرده و خط زیر را اضافه کنید:

```
self.Close()
```

چرا؟

اما تمام این‌ها برای چیست؟ در ابتدا ما باید اطمینان می‌یافتیم که برنامه ما «می‌داند» که ما قصد داریم از دو فرم استفاده کنیم و به همین دلیل بود که با دستور import هر دو فریم FrameMain و FrameSecond را در فایل GUI2 وارد کردیم. پس از آن دوباره با دستور import FrameSecond ارجاعی به فریم دوم را در کدهای FrameMain ایجاد کردیم تا بعدها بتوانیم از آن استفاده کنیم. به عبارت دیگر، می‌خواستیم FrameMain هم از وجود FrameSecond آگاه شود. در بخش _init_ ما برای راحتی کار، متغیری به نام Fs تعریف کردیم که محتوای فریم FrameSecond است. پس از آن و در متد مربوط به کلیک دکمه ShowNew را تعریف کردیم که در صورت کلیک شدن این دکمه، فریم دوم نمایش داده شده و فریم اول یا اصلی مخفی شود. در آخر هم تعیین کردیم که با کلیک دکمه Exit برنامه بسته شود.

حال به سراغ کدهای فریم FrameSecond می‌رویم. تغییراتی که باید در این کدها اعمال شود به نسبت اندک است. در قسمت مربوط به متد __init__ خطی مانند زیر اضافه کنید:

```
self.parent=parent
```

این کار متغیری به نام parent ایجاد می‌کند که به فرم والد FrameSecond (فریمی که آن را صدا زده است) اشاره می‌کند. پس از آن در زیر متد مربوط به رویداد کلیک روی دکمه FSExit دستور event.Skip() را به کامنت تبدیل کرده و دو خط زیر را اضافه کنید:

```
self.parent.Show()
self.Hide()
```

حتماً به خاطر دارید که ما هنگام نمایش FrameSecond فریم اصلی، یعنی FrameMain را مخفی کردیم. به همین دلیل، هنگام بسته شدن FrameSecond باید دوباره FrameMain را (که با نام parent معرفی شده

شکل ۴ افزودن رویداد کلیک به یک دکمه

شکل ۵ ظاهر فریم دوم این برنامه

بود) دوباره به نمایش درآوریم. حال برنامه را اجرا کنید. اگر همه چیز به درستی پیش رفته باشد، شما باید بتوانید با کلیک دکمه btnShowNew فریم اصلی را مخفی و فریم دوم را ظاهر کنید و با کلیک دکمه Exit در فریم دوم، به حالت قبل بازگردید. دکمه Exit در فریم اول هم تمام برنامه را خواهد بست.

قوانین نام‌گذاری

پیش‌تر گفته بودیم که درباره قوانین نام‌گذاری بیشتر صحبت خواهیم کرد. با نام‌گذاری درست کنترل‌های یک برنامه، کد شما خود به خود مستندسازی خواهد شد. به عبارت دیگر، با استفاده از نام‌های درست درک سازوکار برنامه و عملکرد بخش‌های مختلف آن برای سایر «افراد» (استفاده‌کنندگان، ویرایش‌کنندگان یا توسعه دهندگان همکار) ساده‌تر خواهد شد. اگر شما نام‌های پیش‌فرض کنترل‌ها نظیر StaticText1، button1 یا هر چیز دیگری را بپذیرید، در حین ساخت یک فرم پیچیده که شامل تعداد زیادی متن و دکمه و... است، به یقین با مشکل و سردرگمی مواجه خواهید شد. تازه این به شرطی است که شما تنها کسی باشید که با کد سروکار دارید. اگر کسی پس از شما قصد ادامه کار با این کد را داشته باشد، مشکل چندین برابر خواهد شد. بنابراین، پیشنهاد می‌شود که برای برطرف کردن این مشکل از نام‌گذاری دو قسمتی استفاده کنید. بخش اول، باید نوع کنترل را مشخص کند و بخش دوم باید به‌صورت خلاصه عملکرد آن را بیان کند. نام‌هایی که ما در این برنامه استفاده کردیم، نظیر btnExit یا FrameSecond نمونه‌هایی از این سیستم هستند. دو بخش نام‌ها می‌توانند با یک "_" از یکدیگر جدا شوند یا تنها با کوچک و بزرگ نوشتن حروف اول هر بخش. برای کنترل درستی عملکردتان می‌توانید کدهای سه فایل Gui2.py و Framemain.py و FrameSecond.py را از سایت ماهنامه دریافت کنید